A MODIFIED Z-MAP COMPUTATIONAL INTERPOLATION ALGORITHM FOR SURFACE MACHINING SIMULATION

Jo-Peng Tsai[1,2]
[1]*Department of Computer Science & Information Engineering, Far East University, Tainan, Taiwan*
[2]*Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan*
E-mail: perng@cc.feu.edu.tw

ICETI 2012-H1022_SCI
No. 13-CSME-38, E.I.C. Accession 3496

ABSTRACT

The *Z*-Map algorithm is one of the most simplistic methods for NC code simulation in obtaining the machined working surface information; therefore, it is often being applied in NC simulation. In this paper, a modified *Z*-Map computational algorithm was proposed to calculate mesh position and Z coordinates of tool scanning surface according to tool path and initial tool scanning boundary in reducing the long computation time. The proposed algorithm not only can be used to derive straight line and arc machining tool path, but also can be adopted for helical and spline curves. There is no need to simplify the non-linear curves beforehand. The method proposed in this paper provides a feasible reference for decision about the reliability and suitability of each machining step.

Keywords: NC simulation, *Z*-Map, interpolation algorithm.

UN ALGORITHME D'INTERPOLATION Z-MAP MODIFIÉ POUR LA SIMULATION NUMÉRIQUE DANS L'USINAGE DE SURFACE

RÉSUMÉ

L'algorithme *Z*-Map est une méthode des plus simples pour le code de simulation CN afin d'obtenir les informations sur la surface usinée ; par conséquent il est souvent appliqué dans la simulation CN. Dans cette étude, un algorithme numérique *Z*-Map modifié est proposé pour le calcul de la position de maillage et les coordonnées Z de l'outil balayant la surface, selon la trajectoire et les limites du balayage initial de l'outil, afin de réduire la longueur du temps de calcul. L'algorithme proposé, non seulement, peut être utilisé pour la dérivée d'une ligne droite et la trajectoire de machinage en arc de cercle, mais aussi peut être adopté pour les courbes hélicoïdales et splinées. Il n'est pas nécessaire de simplifier les courbes nonlinéaires au préalable. La méthode proposée dans cet ouvrage procure une référence valable pour prendre une décision concernant la fiabilité et la pertinence de chaque étape d'usinage.

Mots-clés : simulation CN ; *Z*-Map ; algorithme d'interpolation.

1. INTRODUCTION

In order to speed-up the product development cycle, manufacturing processes are not allowed to incur any error retarding delivery time; therefore, NC simulation becomes more and more important before field machining task is conducted. Besides error prevention, NC simulation process can be used by an experienced engineer to realize the appropriateness for setting machining parameters, the opportunity in enhancing manufacturing processes, and increase of machining efficiency, etc.

Many studies related to Z-Maps have been conducted. For example, Yau et al. [1] adopted Z-Map model on the presentation of the NC simulation with dynamic errors due to high-speed motion as it has a good balance between efficiency and accuracy. Chung et al. [2] derived a generalized tool-bottom scanning method by simplifying a dual-variable surface into four parametric curves so as to calculate intersected projection Z-Map values. Maeng et al. [3] further simplified a dual-variable surface into a single-variable nonlinear quadric curve, but only considered line rather than taking arcs and curves into account. The Z-Map method projects a flat screen (XY meshes) onto the machined object along the Z-axis. The Z-Map method obtains the machined work piece surface by calculating the intersection of the Z-axis direction vector of the grid point on the XY-plane mesh with the cutter surface along the machining tool path. The main advantages of this method include: (1) calculation is simple, and (2) the machined surface can be obtained accurately. However, it has the following disadvantages: (1) precision of the machined surface and the calculation time are influenced by the number of grid points – the more the grid points, the more of the calculation time and the more the computation resources that are consumed, (2) the simulation effect is poorer at the geometry with steep slope – although a dense grid can be utilized, more computation time and computation resources will be consumed. Lee and Ko [4] proposed an enhanced Z-Map method by refining the boundary geometry while keeping the other portions intact so as to enhance simulation accuracy and reduce computation time. Yun et al. [5] enhanced the accuracy by moving the boundary meshes to approach the real boundary. This can obtain better precision without the need for increasing mesh numbers.

To enhance display quality and speed, Liu et al. [6] suggested recording multiple segmented quadrilaterals in a quad-tree data structure of the work piece surface. After tool processing, the surface enveloped by the tool was re-segmented or nodes were removed to form new polygons in accordance with allowable errors. Hence, the number of quadrilaterals can be reduced without affecting the display quality. Mao et al. [7] employed similar concepts to reallocate the grids of the triangle mesh according to local surface flatness. By doing so, Mao reduced the number of grids and obtained the same display effects. Wei et al. [8] predicted the cutting force in a three-axis ball end milling of a sculptured surface based on the Z-Map method. Fussel et al. [9] also estimated cutting force in a five-axis sculptured surface milling operation.

This paper adopts the Z-Map method for NC cutting toolpath simulation. Since the Z-Map method must change the Z-value of every local machining position, there are two ways in achieving this: (1) find tool scanning XY-plane projection range, then find grid points in the range and calculate new Z-value, as the general practice in the past, this method can use the proposed arc and linear interpolation algorithms with the grid spacing as the interpolation step amount to directly obtain the tool scanning grid locations to save the calculation of the tool scanning position; (2) directly use the proposed arc and linear interpolation algorithms to find out the border curve of the tool starting point and movement as the grid locations and Z-value of the border line of the A′ and B′ areas, as shown in Fig. 1, directly conduct interpolation calculation with the linear or arc processing path of the movement of the tool center, meanwhile, translate border lines and Z-values at each interpolation point location to directly obtain the range and Z-coordinates of all the movements of the tool. This is the most efficient Z-Map algorithm. In addition, when performing NC processing simulation, NC interpretation should be conducted at first to directly simulate based on NC code rather than toolpath. In this way, it can better simulate the actual processing results and confirm whether NC code is wrong or not to more rigorously verify the correctness of NC machining. The Z-Map method's

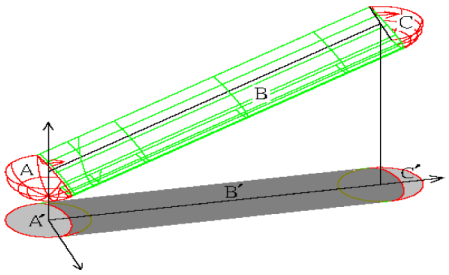

Fig. 1. The cutting tool enveloping surface for the linear cutting of a ball mill. linear cutting of a ball mill.

principles and implementation are as illustrated below.

2. *Z*-MAP METHOD PRINCIPLES

2.1. Toolpath Mesh Interpolation Algorithm

The main principle of applying the *Z*-Map method in NC cutting simulation is to record the *Z*-directional coordinates of the grid points on the *XY* plane corresponding to the enveloping surface formed by the relative movement of the tool surface and workpiece when using the tool for processing. When the whole NC program finishes, each grid point on the *XY* plane will have a record of *Z*-axis height after processing. Then the 3D grid points are connected into triangle meshes to obtain the processed workpiece surface. Therefore, due to the use of different cutting tools and toolpaths (e.g., straight line or arc cutting), the enveloping surfaces of tool scanning will be different. Hence, the intersection programs should be inferred according to the cutting of different tool shapes. In general, the commonly used NC machining tools include flat-end mills, ball-end mills, filleted-end mills and drills, which may result in different forms of enveloping surfaces. Figure 1 illustrates the tool scanning enveloping surface generated by the straight line movement of a ball-end mill. To facilitate calculation, this study divides it into three parts, namely A, B and C, when the projections are on the *XY* plane. Due to the formation of different surfaces in the various parts, they have to be separately calculated. The grid borders of various areas should be calculated before the calculation of the tool surface. As a result, each line of the NC program must be calculated according to the interaction grid and the corresponding *Z*-coordinates. Interpolation calculation is conducted according to grid size using the straight line AC projection and the horizontal movement of the grid interaction for various interpolation grids. Hence, there is no need to calculate the *Z*-values of all grids; in this way, the proposed method eliminates most of the *Z*-value calculation time to make the *Z*-Map method more efficient. The first step of the *Z*-Map method is to calculate the grid of cutting tool at the border intersection on the *XY* plane. In regard to border grid calculation, this study proposes a NC controller-like interpolation algorithm for rapid calculation and the algorithm works described below.

2.1.1. Linear Toolpath Interpolation Algorithm

Using a first quadrant straight line as an example, as shown in Fig. 2, the linear movement from point A to B finishing at a certain point in the middle, *m*, the judgment equation is shown as follows:

$$F_m = Y_m \cdot X_e - X_m \cdot Y_e \tag{1}$$

If $F_m = 0$, the Point *m* is on the line. If $F_m > 0$, the point *m* is above the straight line. When moving one grid (ΔX) to the right, then the judgment equation is

$$F_{m+1} = F_m - Y_e \tag{2}$$

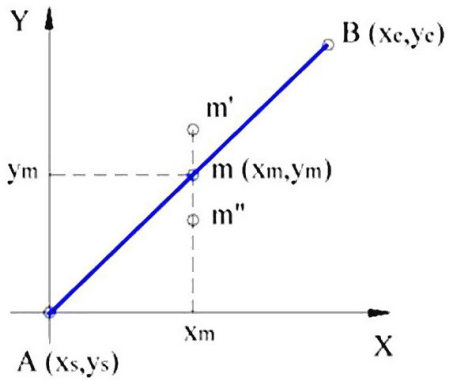

Fig. 2. The first quadrant straight line.

Table 1. Linear border-finding algorithms for various quadrants.

Linear judgment equation	$F_m \geq 0$; $F_{m+1} = F_m - \lvert Y_e \rvert$	$F_m \geq 0$; $F_{m+1} = F_m + \lvert X_e \rvert$
I	$+dx$	$+dy$
II	$-dx$	$+dy$
III	$-dx$	$-dy$
IV	$+dx$	$-dy$

If $F_m < 0$, then point m is below the straight line. Next, when moving one grid (ΔX) upward, then the judgment equation is:

$$F_{m+1} = F_m + X_e \tag{3}$$

Hence, the straight line border grid positioning algorithm of the first quadrant is as follows:

Set $X = X_s$, $Y = Y_s$
BEGIN Evaluate F_m
  WHILE $X <= X_e$ or $Y <= Y_e$ DO:
  BEGIN
    IF $F_m >= 0$ DO:
    BEGIN
    $X = X + \Delta X$
    $F_m = F_m - Y_e$
    END
  ELSE
    BEGIN
    $Y = Y + \Delta Y$
    $F_m = F_m + X_e$
    END
  END
END
END of Algorithm

The methods for computing border grids in the other quadrants are shown in Fig. 3. The judgment equations and movements are summarized as shown in Table 1.

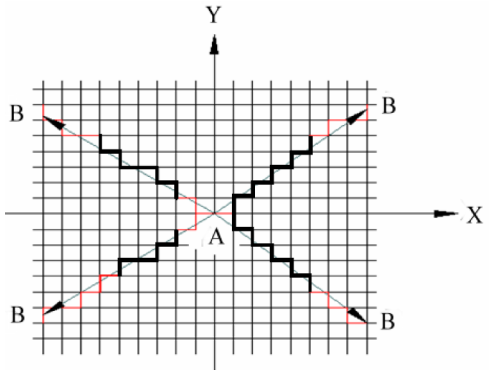

Fig. 3. Movements of border grids in various quadrants.

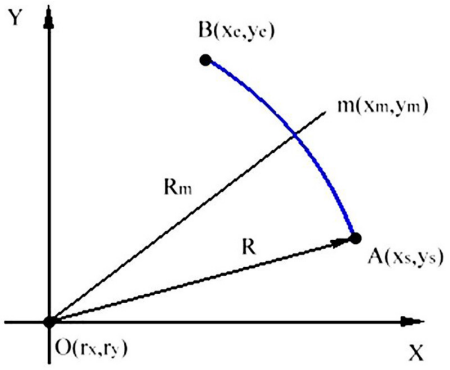

Fig. 4. The first quadrant arc.

2.1.2. Arc Toolpath Interpolation Algorithm

The arc border grid finding methods vary in cases of clockwise and anti-clockwise directions and for different quadrants. Let us take the first quadrant anti-clockwise arc as an example, as shown in Fig. 4; the arc judgment equation is illustrated as

$$F_m = X_m^2 + Y_m^2 - R^2 \tag{4}$$

When $F_m = 0$, then point m is on the round. When $F_m > 0$, the point m is off the round. Next, when moving one grid to the right (ΔX), the F_{m+1} judgment equation of F_{m+1} is

$$F_{m+1} = F_m - 2X_m + 1 \tag{5}$$

When $F_m < 0$, then point m is inside the round. Move upward by one grid (ΔY), and the F_{m+1} judgment equation is

$$F_{m+1} = F_m + 2Y_m + 1 \tag{6}$$

Hence, the border grid positioning algorithm for the anti-clockwise arc in the first quadrant is as follows:

Set $X = X_s$, $Y = Y_s$
BEGIN
Evaluate F_m
WHILE $X <= X_e$ or $Y <= Y_e$ DO:
BEGIN
IF $F_m >= 0$ DO:

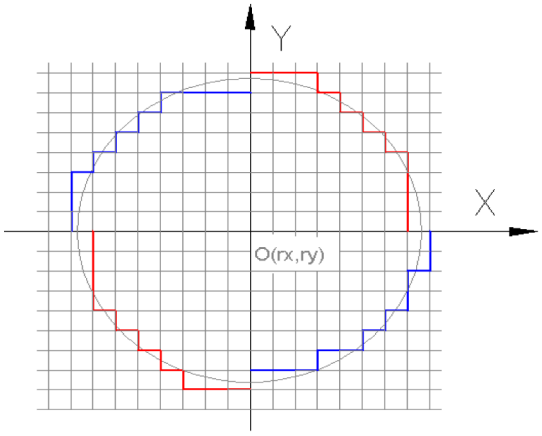

Fig. 5. Arc border-findings for various quadrants.

Table 2. Border-finding algorithms for clockwise and anti-clockwise arcs.

Clockwise	$F_m \geq 0$; $F_{m+1} = F_m - 2Y_m + 1$	$F_m \geq 0$; $F_{m+1} = F_m + 2X_m + 1$	Counter-clockwise	$F_m \geq 0$ $F_{m+1} = F_m - 2X_m + 1$	$F_m \geq 0$ $F_{m+1} = F_m + 2Y_m + 1$
I	$-dy$	$+dx$	I	$-dx$	$+dy$
II	$+dx$	$+dy$	II	$-dy$	$-dx$
III	$+dy$	$-dx$	III	$+dx$	$-dy$
IV	$-dx$	$-dy$	IV	$+dy$	$+dx$

BEGIN
$X = X - \Delta X$
$F_m = F_m - 2X_{m+1}$
END
ELSE
BEGIN
$Y = Y + \Delta Y$
$F_m = F_m + 2Y_{m+1}$
END
END
END
END of Algorithm

The border grid movements of other various quadrants are shown in Fig. 5. The judgment equations and movements are summarized in Table 2.

2.2. Tool Scanning Enveloping Surface Computation Theory

Due to differences in the cutting of tool shapes and processing paths, each line of an NC program produces a different tool enveloping surface. Similarly, different path angles of linear processing produce different tool enveloping surfaces, hence, Z-value conversion of each path should be conducted in the Z-Map method. The ball-end mill and filleted-end mill are inferred as follows.

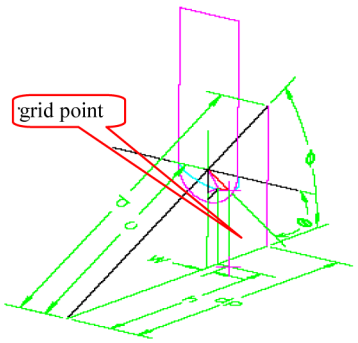

Fig. 6. Ball mill Z-value solution finding diagram.

2.2.1. Ball-end Mill Scanning Enveloping Surface Computation Theory

Figure 1 shows the spherical surfaces generated by Areas A and C when the ball-end mill is rotating in fixed position without movement of the tool scan enveloping surface of the ball mill moving in the linear path. The calculation of the Z-axis coordinate is shown in Fig. 6. When the center of the ball mill is consistent with the starting point (A area), the Z-axis height value of a certain point of the grid, Z_b, is as follows:

$$Z_b = Z_s - \sqrt{R^2 - (w^2 + h^2)} \tag{7}$$

where R in Eq. (7) is the radius of the ball-end mill, w is the vertical distance from grid point to the processing path, and h is the parallel distance from the grid point to the processing path starting point.

Part B shows the tool enveloping surface generated by the tool movements. The computation is relatively complex. However, it is possible to obtain the correction solution directly, according to Choi [10]. The inference process is (see Fig. 6):

$$\theta = \cos^{-1}(w/R) \tag{8}$$

$$\phi = \cos^{-1}(dp/d) \tag{9}$$

$$c = h/\cos\phi - R\sin\theta\tan\phi \tag{10}$$

$$Z_b = z_s + c\sin\phi - R\sin\theta\cos\phi \tag{11}$$

R in Eq. (8) is the radius of the ball-end mill, w is the vertical distance between grid point to the processing path, h in Eq. (10) is the parallel distance between grid point to the processing path starting point. The Z-axis distance of the grid point is the Z_b-value obtained by Eq. (11).

2.2.2. Filleted-end Mill Tool Scanning Enveloping Surface Computation Theory

Similar to the ball-end mill, the calculation of the tool scanning surfaces at the starting point and end point of the path are derived as follows (see Figs. 7, 8):

$$x_t = r_t\cos\theta_2 \tag{12}$$

$$y_t = r_t\sin\theta_2 \tag{13}$$

$$z_t = r_1 - r_1\cos\theta_1 \tag{14}$$

The central part is the tool enveloping curve when the tool moves. To facilitate calculation, a local coordinate is defined, the Z-axis is parallel to the Z-axis of the total coordinates, and the Y-axis is vertical

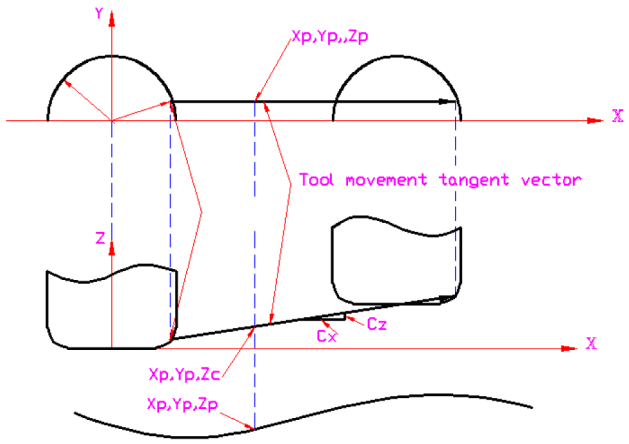

Fig. 7. Diagram of filleted end mill processing in case of local coordinates.

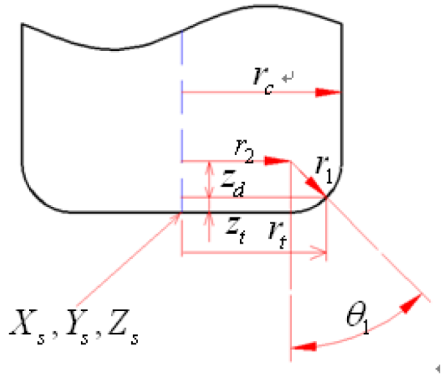

Fig. 8. Sectional diagram of the filleted end mill contact point.

with the tool moving direction. Figure 7 shows the upward view and side view. Figure 8 shows the sectional diagram of the contact point of the filleted end mill. As the tool external surface normal vector **N** is vertical to the tangent vector of the tool motion path **T** [11], hence

$$\mathbf{TN} = 0 \tag{15}$$

$$\mathbf{T} = C_x i + C_z j \tag{16}$$

$$\mathbf{N} = \sin\theta_1 \cos\theta_2 i + \sin\theta_1 \sin\theta_2 j - \cos\theta_1 k \tag{17}$$

$$\therefore \tan\theta_1 \cos\theta_2 = C_z / C_x \tag{18}$$

In Eq. (18), a given θ_1 can be used to obtain θ_2. As shown in Fig. 7, it can map x_t, y_t and determine whether x_t, y_t falls on the grid point for the solution. Hence, using the dichotomy or the Newton method, we can find the corresponding Z-axis coordinate of the grid point.

3. THE APPLICATION OF AN INTERPOLATION ALGORITHM IN THE Z-MAP

As the use of a Z-Map for cutting simulation requires calculating the Z coordinates of all grid points in the range of tool scanning when executing each line of the NC program, the computation sum is very large. As a result, if the coordinates of each grid point have to be calculated by the inferred mathematical program,

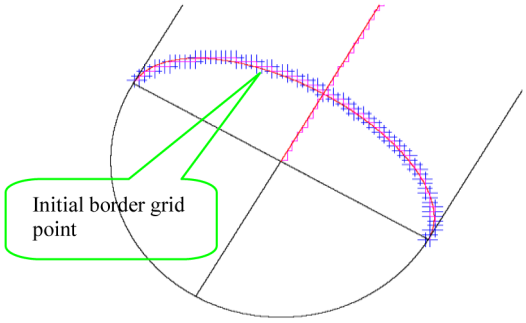

Fig. 9. Initial scanning border of the ball-end mill and the tool path grid point location.

it is time-consuming. Therefore, this paper proposes an interpolation algorithm to simplify the calculation based on the following principles: (1) first calculate the areas generated by the scanning of the tool in fixed positions without movement for each line of the NC program (the starting and ending positions of the tool, see A and C areas in Fig. 1) and the grid point location of the contact border of the tool movement area (Area B) as well as the Z coordinates; (2) then, conduct interpolation calculation of the tool path according to grid size; (3) according to the movement mode of the interpolation movement, calculate the Z coordinates of each movement location in proportion rather than calculating them, according to the inferred equations. Hence, there is no need to calculate the tool scanning border or search for the tool scanning border range. The proposed method can save a huge amount of calculation time. The details are as follows.

3.1. Calculation of the Initial Border of Tool Scanning

The initial border of tool scanning may vary with tool shape and tool path. For example, initial borders differ in cases of ball-end mill in linear cutting, ball-end mill in arc cutting, and filleted-end mill in linear cutting, filleted-end in arc cutting or other tools in different paths. Hence, they should be calculated separately. Next, the commonly used ball-end mill and filleted-end mill are discussed to provide a reference for the calculation of other tools.

3.1.1. Calculation of the Initial Border of the Ball Mill

Using the ball mill and straight line tool paths as shown in Fig. 1 as an example, the calculation method for the grid position of the initial scanning border of the tool is to obtain the borders of Eq. (7), and convert the tangent curve into a straight line and arc by the set tolerance; then apply the straight line path interpolation algorithm from Section 2.1.1 and the arc path interpolation algorithm from 2.1.2 to obtain the initial border grid. Figure 9 illustrates the calculation results of the initial scanning border grid position. After obtaining the border grid point, the Z-axis height is calculated according to the grid position. When the border grid position does not fall on the border line, for the calculation of the Z-axis height Eqs. (8–11) should be applied to obtain the accurate height. Figure 10 shows the calculation results of the Z-axis coordinates of the border.

3.1.2. Calculation of the Filleted-end Mill Scanning Border

Using the filleted-end mill and straight line mill tool paths, as shown in Fig. 11, as an example, the method for the calculation of the grid point locations of the initial sweeping borders of the tool is similar to the method used for the ball mill; however, it uses Eq. (13). The procedures are as follows: (1) set θ_2 to calculate the corresponding θ_1 by 0–180 degrees to obtain the projection lines on X, Y of the border lines; (2) convert the line projection curve into the straight line and arc according to the set tolerance; (3) apply

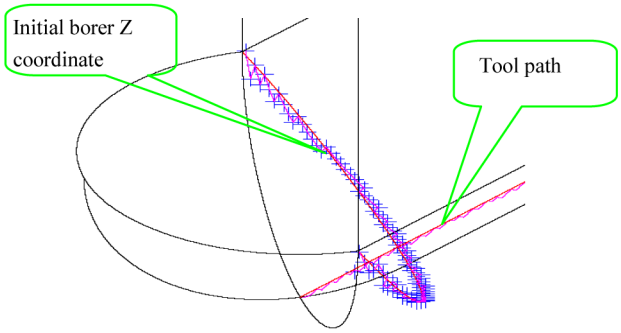

Fig. 10. The initial border Z-value of the ball-end mill and the tool path grid point position.

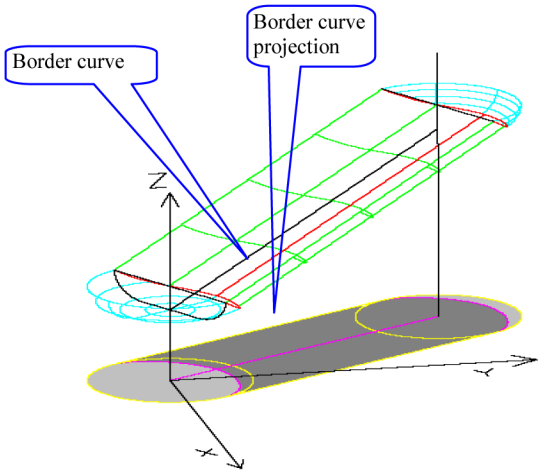

Fig. 11. Border curve of the filleted end mill in straight line processing.

the straight line interpolation algorithm as shown in Section 2.1.1 and the arc path interpolation algorithm from Section 2.1.2 to obtain the initial border grid. After obtaining the border grid point, the Z-axis height is calculated according to the grid position. The calculation method is to use Eqs. (15–18) and obtain the correct Z-axis height positions by the numerical method. Figure 12 shows the calculation results of the Z-axis coordinates of the border.

3.2. Application of the Path Interpolation Algorithm in the Calculation of Tool Scanning

To save time in the calculation of the Z-axis coordinates of the overall tool path scanning, the scanning curve of the tool processing is regarded as the overall tool scanning surface at the border curve position according to the movement of each grid point of the tool path interpolation value. This concept resembles the movement processing along the interpolation value position grid of the processing path of the tool. This practice is different from the actual processing, and it is conducted according to the grid errors. To obtain a better accuracy rate, the mesh should be made smaller. Figure 13 illustrates the calculation results of the filleted end mill as shown in Fig. 12 and the straight line processing path. The filleted-end mill with a diameter of 20 mm and a reverse angle of R5 mm is used. The grid accuracy is 0.3 mm. Figure 14 shows the Z-Map processing simulation results as shown in Fig. 13.

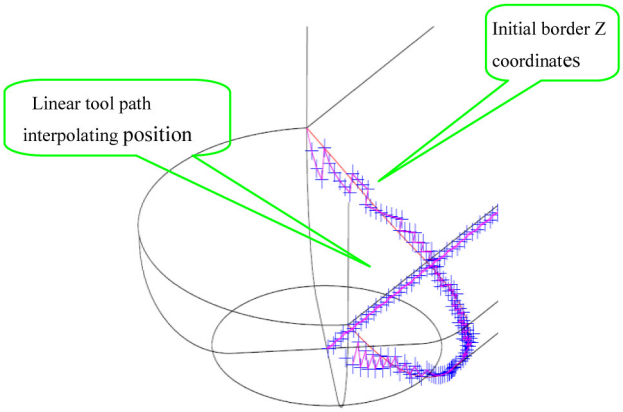

Fig. 12. Z-value of the filleted end mill's border and linear processing path grid point.

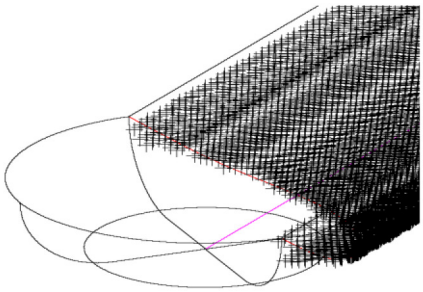

Fig. 13. The calculation results of straight line processing for the filleted end mill by using the interpolation algorithm.

3.3. Application of Cutting Simulation

In this paper VB6 language is used to develop and test the system on a PC of Pentium IV 1.7 GHz with a Microsoft operating system. Figure 15 shows the processing path for a back mirror cavity sized at 160 mm $\times$124 mm $\times$ 50 mm. A flat-end mill diameter 20 mm is used to process the work piece by single pass in depth of 1.5 mm. The Z-Map simulated grid accuracy is 0.3 mm. Figure 16 shows the simulation results, and the calculation time is about 48 seconds.

4. DISCUSSION

A helical curve can be represented by an arc along with its z coordinate and a spline curve can be broken into many piecewise lines; therefore, the proposed algorithm in this paper can not only be used to derive a straight line and arc machining tool path, but it can also be adopted for helical and spline curves.

The specifications of the adopted computer in this paper, the size of the workpiece, the mesh size, and the computation time have been illustrated in Section 3.3. However, it still needs to further verify the effectiveness in comparison with other researches. For the flat-end and the ball-end tools, they are the exact solutions which are directly solved by equations. As for the filleted end milling tool, the permitted tolerance can be set in advance for the numerical solutions of nodes. Therefore, the proposed algorithm can be successfully applied in flat-end, ball-end, and filleted end milling tools for the verification of correctness in the development of an NC simulation system.

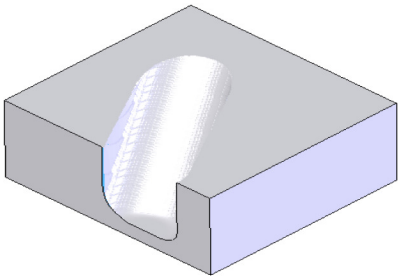

Fig. 14. The simulation results of *Z*-Map processing when using the interpolation algorithm to calculate straight line processing of the filleted end mill.

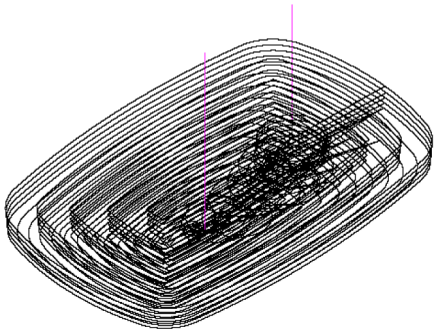

Fig. 15. Tool path of cutting back mirror cavity, using diameter 20 mm end mill.

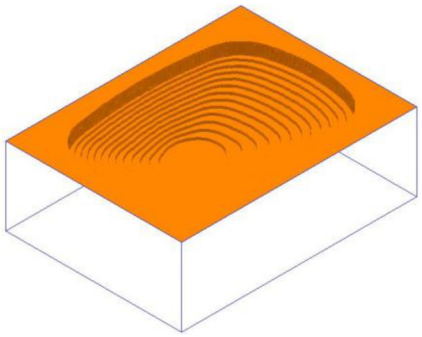

Fig. 16. The simulation results of back mirror die with rough cutting, using flat-end mill.

5. CONCLUSIONS

Previous studies of the *Z*-Map method did not explore the method of grid location. This paper proposes to effectively apply the grid location method in the *Z*-Map method grid location and tool scanning surface *Z*-axis coordinate calculation. The process is very simple in logic and calculation as it is time-efficient in the simulation of each line of NC path. Although this paper discusses only the linear and arc border positioning algorithms, the calculation of curves can be converted into the calculation of straight line and arc. Hence, the method can be applied in straight line and arc path simulations as well as curve processing path simulations.

ACKNOWLEDGEMENTS

The author appreciates the grant support from the Precision Machinery Research and Development Center.

REFERENCES

1. Yau, H.T., Ting, J.Y. and Chuang, C.M., "NURBS machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces", *International Journal of Advanced Manufacturing Technology*, Vol. 23, No. 7–8, pp. 577–585, 2004.
2. Chung, Y.C., Park, J.W., Shin, H. and Choi, B.K., "Modeling the surface swept by a generalized cutter for NC verification", *Computer Aided Design*, Vol. 30, No. 8, pp. 587–594, 1998.
3. Maeng, S.R., Baek, N., Shin, S.Y. and Choi B.K., "A Z-map update method for linearly moving tools", *Computer Aided Design*, Vol. 35, No. 11, pp. 995–1009, 2003.
4. Lee, S.K. and Ko, S.L., "Development of system for machining process using enhanced Z-map model", *Journal of Material Processing Technology*, Vol. 130–131, pp. 608–617, 2002.
5. Yun, W.S., Ko, J.H., Lee, H.U., Cho, D.W. and Ehmann, K F., "Development of a virtual machining system, Part 3: Cutting process simulation in transient cuts", *International Journal of Machine Tools & Manufacture*, Vol. 42, No. 15, pp. 1617–1626, 2002.
6. Liu, S.Q., Ong, S.K., Chen, Y.P. and Nee, A.Y.C., "Real-time, dynamic level-of-detail management for three-axis NC milling simulation", *Computer-Aided Design*, Vol. 38, No. 4, pp. 378–391, 2006.
7. Mao, J., Liu, S. and Gao, Z., "Three-axis NC milling simulation based on adaptive triangular mesh", *Computers & Industrial Engineering*, Vol. 60, No. 1, pp. 1–6, 2011.
8. Wei, Z.C., Wang, M.J., Zhu, J.N. And Gu, L.Y., "Cutting force prediction in ball end milling of sculptured surface with Z-level contouring tool path", *International Journal of Machine Tools & Manufacture*, Vol. 51, No. 5, pp. 428–432, 2011.
9. Fussell, B.K., Jerard, R.B. and Hemmett, J. G., "Modeling of cutting geometry and force for 5-axis sculptured surface machining", *Computer Aided Design*, Vol. 35, No. 4, pp. 333–346, 2003.
10. Choi, B.K., *Surface Modeling for CAD/CAM*, pp. 354–356, Elsevier Science Publisher, 1991.